A SIMPLE APPROXIMATION FOR THE THROUGHPUT OF TANDEM KANBAN LINES

Matthew F. Keblis
and
Izak Duenyas
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 93-38

December 1993

A SIMPLE APPROXIMATION FOR THE THROUGHPUT OF TANDEM KANBAN LINES

MATTHEW F. KEBLIS and IZAK DUENYAS
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, Michigan 48109

Abstract

We consider a production system consisting of a tandem line with machines having general processing time distributions. Releases to the system are governed by the *kanban* release mechanism. We derive a simple approximation for this system. Comparisons with simulation show our approximation to be robust over a wide range of conditions.

1 Introduction

In the last decade, a great deal of research activity has focused on release control in tandem manufacturing systems. Release control in tandem systems is important as a mechanism to control WIP costs and cycle times in a production facility. In many manufacturing systems, the method for controlling releases is MRP, in which releases are scheduled by subtracting fixed lead times from due dates. Pull systems such as kanban (Hall (1983), Monden (1983) and Ohno (1988)), have recently become very popular due to the success of Japanese just-in-time methods.

A variety of researchers have developed models of tandem systems under pull release mechanisms. Kimura and Terada (1981) developed a set of structural equations to explore system behavior under various conditions. Karmarkar and Kekre (1986) studied batching policies in two-cell Markovian systems and investigated the effect of batch size on expected inventory and backorder costs. Buzacott (1989) developed a general model of a tandem production line and showed that both MRP and kanban systems can be obtained as special cases. Deleersnyder et al. (1989) developed a discrete-time Markov model of a kanban system and then studied the effect of machine reliability and demand fluctuation on throughput. Mitra and Mitrani (1990) developed an approximation procedure to calculate the throughput of a tandem kanban line where all the machines have exponential processing times. Their model analyzes each cell in isolation and links them together using a set of fixed point equations. The set of fixed-point equations is then solved using an iterative approach. In Mitra and Mitrani (1991) this model is extended to the case of stochastic demands. Spearman (1992) considers the issue of customer service in both kanban and CONWIP systems. He investigates the effects of changing inventory levels and processing time characteristics on customer service. Buzacott et al. (1992) also investigate the issue of service level provided to customers in kanban systems. Tayur (1992) shows that certain allocations of kanbans are superior to others in tandem lines. A recent summary of the models of kanban-based pull systems can be found in Uzsoy and Martin-Vega (1990).

Despite the great interest in kanban, most models for even the simplest tandem kanban lines have rather restrictive assumptions, such as exponential processing times. This is in contrast to tandem production lines under the CONWIP release mechanism, which can be modeled as closed queueing networks. Robust approximations have been developed for closed queueing networks with general processing times (e.g., Shanthikumar and Gocmen (1983), Whitt (1984)).

Under the CONWIP release mechanism (See Spearman et al. (1990) and Spearman and Zazanis (1992) for discussions of CONWIP), the release of work into the system is controlled by cards as in kanban. The total number of cards in a line is held constant. Each time a job is completed, its card is removed and sent to the front of the line to authorize the start of a new job. Once jobs are released to the first machine in each line, they are then "pushed" through the system.

When a card arrives to the first machine, a new job is not necessarily released to the system immediately. The card only authorizes the operator at the first machine to release a new job when she is done processing the job she is working on. This is due to the fact that releasing a new job before the operator at the first machine is ready to work on it would only lead to the job waiting at a queue in front of machine 1. Therefore, in a CONWIP system, the first machine in the line would have at most 1 unit of WIP being worked on, although at any time, there may be several cards unattached to jobs at the machine.

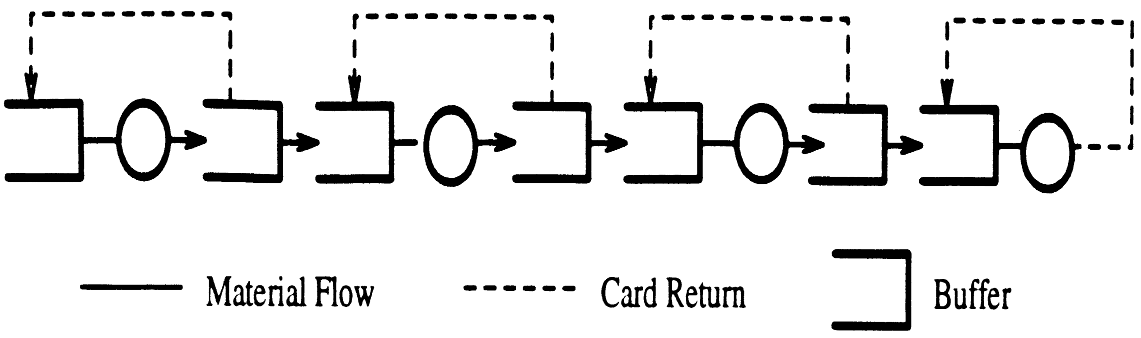

Figure 1: Tandem Kanban Line

An alternative to the CONWIP release mechanism described above is to use the *kanban* release mechanism. In a kanban system, each machine is allocated one or more cards. If there is an unattached card on the bulletin board of the first machine in the line, a new job is released and the card is attached to that job. Once this job's processing is finished on the first machine, the job moves to the input buffer of the next machine if there is a card (unattached to any job) on the bulletin board of the second machine. In this case, the card from machine 1 that was attached to the job is detached and sent back to the bulletin board of the first machine and the card on the bulletin board of the second machine is attached to the job. Otherwise, the job waits in the output hopper of the first machine until a card becomes available at the second machine. The movement of the jobs through the other machines in the line follows the same procedure.

In this paper, we focus on systems under the kanban release mechanism. Our focus on these systems is motivated by their prevalence in industry and the fact that practitioners mainly use simulation to predict the performance of these systems. This paper provides a simple, robust approximation for the throughput of these systems. To derive an approximation for the throughput of a tandem line under the kanban release mechanism, we make use of a state space approximation first introduced by Akyildiz (1988) to approximate the throughput of closed queueing networks with blocking. The state space approximation replaces the kanban system by an approximately equivalent CONWIP system. We then use known results for approximating the throughput of CONWIP systems to obtain an approximation for the throughput of a system under the kanban release mechanism.

The remainder of this paper is organized as follows. In the next section, we introduce the notation and problem formulation. Section 3 describes our approximation. In Section 4, we test the performance of our approximation against simulation results. Section 5 concludes the paper.

2 Problem Formulation

We consider a production line with m machines as shown in Figure 1. The i^{th} machine, has n_i kanban cards allocated to it. The movement of the jobs in the line is according to the *kanban* discipline as described above. Raw material is always assumed to be available at the first station. We assume that successive processing times at machine i are independent with finite mean, x_i and variance σ_i^2.

When a job is finished at the last machine in the production line, an output occurs, and the card attached to that job is sent back to the bulletin board of the last machine. We define N_t as the number of outputs until time t, starting from time 0. We are interested in finding the throughput, $\theta = \lim_{t\to\infty} N_t/t$.

3 An Approximation for Throughput

Our approximation for the throughput of a tandem kanban system is based on a state-space approximation developed by Akyildiz (1988). In Akyildiz (1988), the state space approximation was used to approximate the throughput of a closed queueing network with blocking with the throughput of an approximately equivalent closed queueing network without blocking. In this paper, we use a similar idea to approximate the throughput of tandem lines under the kanban release mechanism by tandem lines under CONWIP release.

In this section, we derive an approximation for a tandem kanban system with M stations. Machine i is assumed to have n_i cards allocated to it. In order to compute an approximation, we need to compute the size of the state space of a tandem kanban system, where the state space consists of all the possible ways in which jobs can be distributed in the system, given the card counts for each station. As previously noted in Tayur (1992), the distribution of jobs in a tandem kanban system with M stations can be completely characterized by an $M-1$-vector where the j^{th} element of the vector represents the difference between the number of jobs in the output hopper of cell j, and the number of unattached cards in the bulletin board of cell $j+1$. For example, in a 2-station tandem kanban line with n_1 cards in station 1 and n_2 cards in station 2, the state of the system can be described by a single number from the set $\{n_1, n_1-1, \ldots, -n_2\}$. Therefore, the state space is of size n_2+n_1+1. Note that when the state of the system is n_1, n_1 jobs are in the output hopper of station 1, and there are no unattached cards in station 2 (therefore n_2 jobs are waiting to be processed in station 2), and station 1 is idle. Similarly, the state $-n_2$ corresponds to the case where there are no jobs in station 1's output hopper and n_2 unattached cards on the bulletin board of station 2. The size of the state space of a tandem kanban system can be calculated by the following recursion where Y_1 is the number of states (Tayur 1992).

$$\begin{aligned}
X_M &= n_M + 1 \\
Y_M &= 1 \\
X_{m-1} &= (n_{m-1} + 1)X_m + (n_{m-1}(n_{m-1} + 1)/2)Y_m \qquad \text{m=M,...,2} \\
Y_{m-1} &= X_m + n_{m-1}Y_m
\end{aligned}$$

We also note the well known fact that the number of ways N jobs can be distributed in a closed queueing network with M machines is given by $\begin{pmatrix} M+N-1 \\ M-1 \end{pmatrix}$. Therefore, since a tandem CONWIP system can be represented by a closed queueing network, this expression also gives the size of the state space for a tandem CONWIP system.

Given the above expressions for the size of the state space in tandem CONWIP and kanban systems, we first note that the number of states in a tandem CONWIP system with 2 machines and N cards is given by $N + 1$. As we previously stated, the size of the state space in a 2-machine kanban system is $n_1 + n_2 + 1$. Therefore, if $n_1 + n_2 = N$, then the size of the state space for kanban and CONWIP tandem systems is the same. It is easy to show that a 2-machine kanban system with $n_1 + n_2 = N$ total cards has the same throughput as a 2-machine CONWIP system with N cards. Therefore, for a 2-machine system, the kanban and CONWIP release mechanisms result in the same size state space, and the same throughput. It is rather interesting to note that Akyildiz (1988) similarly points out that for each 2-station closed queueing network with blocking, there exists a 2-station closed queueing network without blocking and fewer jobs such that the two networks have the same throughput and the same number of states.

For a kanban system with more than 2 machines, a tandem CONWIP system that has the same number of states as a tandem kanban system does not necessarily exist. Let the size of the state space of the kanban system, as computed by the recursion above be given by S_{kan}. For the same system with CONWIP release, let N_1 be the largest card count resulting in a tandem CONWIP system with fewer states than S_{kan}.

We let $S_C(N_1)$ be the number of states for this CONWIP system (hence, $S_C(N_1) < S_{kan} \leq S_C(N_1+1)$) and $\theta_C(N_1)$ represent the throughput of this CONWIP system. Our throughput approximation for the tandem kanban system is then given by a linear interpolation of the throughput of the CONWIP systems with N_1 and $N_1 + 1$ cards. Letting θ_{ap} represent the approximation for the throughput of the kanban system, we get

$$\theta_{ap} = \theta_C(N_1) + \frac{(S_{kan} - S_C(N_1))(\theta_C(N_1+1) - \theta_C(N_1))}{S_C(N_1+1) - S_C(N_1)}. \tag{1}$$

The remaining issue in our approximation is the throughput of the CONWIP system, $\theta_C(N_1)$. Clearly, if all the processing times are exponential, then the throughput of the tandem CONWIP system can be computed exactly, by using Mean Value Analysis (MVA) (Reiser and Lavenberg, 1980). However, exact results for the throughput of closed queueing networks with general processing times do not exist. Therefore, when processing times are general, we use an approximation due to Shanthikumar and Gocmen (1983) for the throughput of a closed queueing network. This approximation is very simple to code and has been shown to be robust for a wide variety of cases when processing times at each station have coefficients of variation less than 1. Since, in almost all practical situations, processing time distributions encountered in manufacturing satisfy this condition, we make use of this approximation by Shanthikumar and Gocmen. Despite the fact that we are using approximations at two levels, the results in Section 4 show that our approximation is robust.

4 Computational Results

In this section, we report the results of our simulation study in which we tested the performance of our approximation. We used a GPSS/H program to simulate the tandem kanban lines. Each simulation run lasted 21000 time units, and we initialized the system after the first 1000 time units to account for initial bias. The average of 10 runs gave us our simulation estimate for the throughput. Whereas each simulation run took several minutes on a Sun SPARCstation 10 machine, our approximation always gave results in less than a second. Below, we report some representative examples to demonstrate the performance of our approximation.

We display eighty cases as examples of our experience with the approximation. Tables 1 and 2 contain cases where the production line consists of four cells and each cell contains a single machine with an exponential processing time distribution. In Table 1, all the machines have mean processing times equal to 0.25. In Table 2, the mean processing times of the machines are varied between 0.25 and 1.00. The same kanban card allocations are considered in both tables. Tables 3 and 4 contain cases where the production line consists of eight cells and each cell contains a single machine with an exponential processing time distribution. In Table 3, all the machines have mean processing times equal to 0.25. In Table 4, the mean processing times of the machines are varied between 0.25 and 1.00. The same kanban card allocations are considered in both tables. Tables 5 and 6 contain cases where the production line consists of four cells and each cell contains a single machine with a general processing time distribution. In Table 5, all the machines have Erlang-2 processing time distributions with mean equal to 0.25. In Table 6, each machine has one of the following processing time distributions: exponential, Erlang-2, Erlang-3 or Erlang-4. The mean processing times of these machines are varied between 0.25 and 1.00. The same kanban card allocations are considered in both tables. Tables 7 and 8 contain cases where the production line consists of eight cells and each cell contains a single machine with a general processing time distribution. In Table 7, all the machines have Erlang-2 processing time distributions with mean equal to 0.25. In Table 8, each machine has one of the following processing time distributions: exponential, Erlang-2, Erlang-3 or Erlang-4. The mean processing times of these machines are varied between 0.25 and 1.00. The same kanban card allocations are considered in both tables.

The results for all eighty cases are in Tables 1-8. In each table, the first column gives the allocation of cards to machines. For example, the first allocation in Table 1 (1,2,1,1) means that the four machines in the production line were allocated, respectively 1,2,1 and 1 cards. The second column gives the mean processing time of each machine in the line. In Tables 6 and 8, the last column gives the processing time distribution for each machine not having an Erlang-2 distribution. In these tables, θ_s denotes the throughput obtained by simulation, θ_{ap} denotes the throughput obtained by our approximation, while θ_f denotes the throughput obtained from the fixed-point approximation of Mitra and Mitrani. Finally, we also give the percentage difference between the simulation results and the results of both approximations.

As it can be seen from Tables 1-8, our approximation behaved very well for all of the test problems considered. This includes systems with exponential processing times, as well as systems with general processing times. The maximum error, in all cases considered, was less than 4 %, and the average error was much lower than that. The approximation behaved very well for different values of coefficient of

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	%err	θ_f	%err
1	1,2,1,1	$x_1..x_4$=0.25	2.305	2.286	-0.8	2.272	-1.4
2	1,2,2,1	$x_1..x_4$=0.25	2.540	2.500	-1.6	2.499	-1.6
3	1,3,2,1	$x_1..x_4$=0.25	2.679	2.649	-1.1	2.642	-1.4
4	2,2,2,2	$x_1..x_4$=0.25	2.750	2.707	-1.5	2.722	-1.0
5	1,3,3,1	$x_1..x_4$=0.25	2.817	2.781	-1.3	2.778	-1.4
6	1,8,7,1	$x_1..x_4$=0.25	3.356	3.338	-0.5	3.306	-1.5
7	1,7,7,1	$x_1..x_4$=0.25	3.326	3.303	-0.7	3.274	-1.6
8	1,17,18,1	$x_1..x_4$=0.25	3.679	3.670	-0.2	3.627	-1.4
9	2,3,3,2	$x_1..x_4$=0.25	2.975	2.934	-1.4	2.949	-0.9
10	1,4,4,1	$x_1..x_4$=0.25	3.006	2.974	-1.1	2.962	-1.5

Table 1: Examples 1-10

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	%err	θ_f	%err
11	1,2,1,1	x_1 = 1.00, $x_2..x_4$=0.25	0.975	0.975	0.0	0.9819	0.7
12	1,2,2,1	$x_1..x_3$=0.25, x_4=1.00	0.987	0.991	0.4	0.987	0.0
13	1,3,2,1	x_1 = x_4=0.50, x_2 = x_3=0.25	1.645	1.609	-2.1	1.672	1.6
14	2,2,2,2	x_1 = x_4=0.25, x_2 = x_3=0.50	1.581	1.635	3.4	1.573	-0.5
15	1,3,3,1	x_1=0.25, x_2=0.33, x_3=0.50, x_4=1.00	0.968	0.988	2.0	0.967	-0.1
16	1,8,7,1	x_1=1.00, x_2=0.50, x_3=0.33, x_4=0.25	0.994	1.000	0.5	0.999	0.5
17	1,7,7,1	x_1 = x_2=0.50, x_3 = x_4=0.25	1.777	1.849	4.0	1.778	0.0̃
18	1,17,18,1	x_1 = x_3 = x_4=0.25, x_2=0.50	2.004	2.000	-0.2	1.999	-0.2
19	2,3,3,2	x_1 = x_2 = x_4=0.25, x_3=0.33	2.660	2.631	-1.1	2.623	-1.4
20	1,4,4,1	x_1 = x_3=0.33, x_2 = x_4=0.25	2.522	2.474	-1.9	2.475	-1.9

Table 2: Examples 11-20

variation of processing time and also for different kanban card allocations. In terms of computational time, we must note that our approximation produced results much more rapidly than the Mitra and Mitrani approximation which became considerably slower as the number of cards in the system was increased. In Example 7 the CPU time using the Mitra and Mitrani approximation was 77 seconds. When the same system was examined with the card allocation (1,17,18,1), i.e. Example 8, the CPU time was 5 hours. In the case of our approximation, the CPU time was less than 1 second in both examples.

5 Conclusions and Further Research

In this paper, we derived an approximation for the throughput of tandem kanban lines. The approximation is based on a state-space approximation originally developed by Akyildiz. We tested our approximation on a variety of examples and found that the approximation works very well. Among the advantages

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	%err	θ_f	%err
21	1,2,3,4,4,3,2,1	$x_1..x_8$=0.25	2.675	2.678	0.1	2.657	-0.7
22	1,2,4,4,4,4,2,1	$x_1..x_8$=0.25	2.740	2.761	0.8	2.729	-0.4
23	1,2,2,2,1,1,1,1	$x_1..x_8$=0.25	1.996	2.046	2.5	1.936	-3.0
24	2,2,2,2,2,2,2,2	$x_1..x_8$=0.25	2.505	2.448	-2.3	2.451	-2.2
25	1,2,2,1,1,2,2,1	$x_1..x_8$=0.25	2.173	2.158	-0.7	2.092	-3.7
26	3,3,3,3,4,4,4,4	$x_1..x_8$=0.25	2.973	2.950	-0.8	2.943	-1.0
27	1,3,3,3,4,4,4,1	$x_1..x_8$=0.25	2.853	2.825	-1.0	2.811	-1.5
28	1,1,2,2,2,2,1,1	$x_1..x_8$=0.25	2.203	2.159	-2.0	2.149	-2.5
29	1,1,1,2,2,1,1,1	$x_1..x_8$=0.25	1.991	1.945	-2.3	1.925	-3.3
30	2,2,2,2,2,2,2,1	$x_1..x_8$=0.25	2.458	2.394	-2.6	2.409	-2.0

Table 3: Examples 21-30

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	%err	θ_f	%err
31	1,2,3,4,4,3,2,1	x_1=1.00, $x_2..x_8$=0.25	0.983	1.000	1.7	0.988	0.5
32	1,2,4,4,4,4,2,1	$x_1..x_7$=0.25, x_8=1.00	0.986	1.000	1.4	0.988	0.2
33	1,2,2,2,1,1,1,1	$x_1 = x_8$=0.50, $x_2..x_7$=0.25	1.515	1.496	-1.3	1.527	0.8
34	2,2,2,2,2,2,2,2	$x_1..x_4$=0.25, $x_5..x_8$=0.50	1.714	1.699	-0.9	1.755	-2.3
35	1,2,2,1,1,2,2,1	$x_1 = x_8$=1.00, x_2=0.33, $x_3..x_7$=0.25	0.893	0.858	-3.9	0.974	9.0
36	3,3,3,3,4,4,4,4	x_1=1.00, x_2=0.50, x_3=0.33, $x_4..x_8$=0.25	0.986	1.000	1.4	0.992	0.6
37	1,3,3,3,4,4,4,1	$x_1..x_6$=0.25, $x_7 = x_8$=0.33	2.491	2.516	1.0	2.466	-1.0
38	1,1,2,2,2,2,1,1	$x_1..x_6$=0.25, $x_7 = x_8$=0.33	1.910	1.963	2.8	1.876	-1.8
39	1,1,1,2,2,1,1,1	$x_1 = x_2 = x_4..x_8$=0.25, x_3=0.33	1.878	1.852	-1.4	1.809	-3.7
40	2,2,2,2,2,2,2,1	$x_1 = x_3$=0.33, $x_2 = x_4..x_8$=0.25	2.237	2.165	-3.2	2.197	-1.8

Table 4: Examples 31-40

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	%err
41	1,2,1,1	$x_1..x_4$=0.25	2.669	2.683	0.5
42	1,2,2,1	$x_1..x_4$=0.25	2.940	2.916	-0.8
43	1,3,2,1	$x_1..x_4$=0.25	3.071	3.063	-2.5
44	2,2,2,2	$x_1..x_4$=0.25	3.150	3.118	-1.0
45	1,3,3,1	$x_1..x_4$=0.25	3.204	3.186	-0.6
46	1,8,7,1	$x_1..x_4$=0.25	3.625	3.616	-0.2
47	1,7,7,1	$x_1..x_4$=0.25	3.603	3.592	-0.3
48	1,17,18,1	$x_1..x_4$=0.25	3.829	3.823	-0.2
49	2,3,3,2	$x_1..x_4$=0.25	3.344	3.316	-0.8
50	1,4,4,1	$x_1..x_4$=0.25	3.363	3.348	-0.4

Table 5: Examples 41-50

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	$\%err$	Distribution
51	1,2,1,1	$x_1 = 1.00$, $x_2..x_4=0.25$	0.984	0.983	-0.1	$1 = EX$
52	1,2,2,1	$x_1..x_3=0.25$, $x_4=1.00$	0.993	0.995	0.2	$4 = EX$
53	1,3,2,1	$x_1 = x_4=0.50$, $x_2 = x_3=0.25$	1.756	1.728	-1.6	$1 = E3$, $4 = EX$
54	2,2,2,2	$x_1 = x_4=0.25$, $x_2 = x_3=0.50$	1.687	1.748	3.6	$2 = EX$, $3 = E3$
55	1,3,3,1	$x_1=0.25$, $x_2=0.33$, $x_3=0.50$, $x_4=1.00$	1.002	1.000	-0.2	$2 = EX$, $3 = E3$, $4 = E4$
56	1,8,7,1	$x_1=1.00$, $x_2=0.50$, $x_3=0.33$, $x_4=0.25$	0.995	1.000	0.5	$1 = EX$, $2 = E3$, $3 = E4$
57	1,7,7,1	$x_1 = x_2=0.50$, $x_3 = x_4=0.25$	1.918	1.945	1.4	$1 = 2 = E3$
58	1,17,18,1	$x_1 = x_3 = x_4=0.25$, $x_2=0.33$	3.010	3.003	-0.2	$2 = E3$
59	2,3,3,2	$x_1 = x_2 = x_4=0.25$, $x_3=0.33$	2.931	2.941	0.4	$3 = E3$
60	1,4,4,1	$x_1 = x_3=0.33$, $x_2 = x_4=0.25$	2.850	2.849	0.0	$1 = 3 = E4$

Table 6: Examples 51-60

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	$\%err$
61	1,2,3,4,4,3,2,1	$x_1..x_8=0.25$	3.086	3.101	0.5
62	1,2,4,4,4,4,2,1	$x_1..x_8=0.25$	3.137	3.174	1.2
63	1,2,2,2,1,1,1,1	$x_1..x_8=0.25$	2.361	2.437	3.2
64	2,2,2,2,2,2,2,2	$x_1..x_8=0.25$	2.960	2.880	-2.7
65	1,2,2,1,1,2,2,1	$x_1..x_8=0.25$	2.590	2.568	-0.9
66	3,3,3,3,4,4,4,4	$x_1..x_8=0.25$	3.362	3.333	-0.9
67	1,3,3,3,4,4,4,1	$x_1..x_8=0.25$	3.259	3.230	-0.9
68	1,1,2,2,2,2,1,1	$x_1..x_8=0.25$	2.605	2.568	-1.4
69	1,1,1,2,2,1,1,1	$x_1..x_8=0.25$	2.375	2.313	-2.6
70	2,2,2,2,2,2,2,1	$x_1..x_8=0.25$	2.908	2.855	-1.8

Table 7: Examples 61-70

Example	Kanban Allocation	Mean Processing Time	θ_s	θ_{ap}	$\%err$	Distribution
71	1,2,3,4,4,3,2,1	$x_1=1.00$, $x_2..x_8=0.25$	0.987	1.000	1.3	$1 = EX$
72	1,2,4,4,4,4,2,1	$x_1..x_7=0.25$, $x_8=1.00$	1.001	1.000	-0.1	$4 = EX$
73	1,2,2,2,1,1,1,1	$x_1 = x_8=0.50$, $x_2..x_7=0.25$	1.644	1.667	1.4	$1 = E4$, $8 = EX$
74	2,2,2,2,2,2,2,2	$x_1..x_4=0.25$, $x_5..x_8=0.50$	1.732	1.678	-3.1	$3 = E4$
75	1,2,2,1,1,2,2,1	$x_1 = x_8=1.00$, $x_2=0.33$, $x_3..x_7=0.25$	0.966	0.953	-1.4	$1 = 2 = E3$, $8 = E4$
76	3,3,3,3,4,4,4,4	$x_1 = x_2=0.33$, $x_3..x_8=0.25$	2.849	2.899	1.8	$1 = 2 = E4$
77	1,3,3,3,4,4,4,1	$x_1..x_8=0.33$	2.446	2.440	-0.3	$1..8 = E2$
78	1,1,2,2,2,2,1,1	$x_1..x_8=0.33$	1.955	1.928	-1.4	$1..8 = E2$
79	1,1,1,2,2,1,1,1	$x_1..x_6=0.25$, $x_7 = x_8=0.33$	2.034	2.105	3.5	$7 = 8 = E2$
80	2,2,2,2,2,2,2,1	$x_1 = x_3=0.33$, $x_2 = x_4..x_8=0.25$	2.661	2.614	-1.8	$1 = 3 = E3$

Table 8: Examples 71-80

of this approximation is that unlike most other approximations for kanban systems, it does not require exponential processing times. In addition, since the approximation is not computationally expensive, it can be used for the design of tandem kanban lines. As the kanban release mechanism is used widely in industry, our approximation should aid manufacturers in analyzing and designing their production systems.

Further research should focus on more complex systems and performance measures other than throughput. In Duenyas and Keblis (1993) we extend our approximation to the case of an assembly system where several subassembly lines feed an assembly station. We also compare the performance of different release mechanisms. However further research is necessary to develop approximations for more complex systems.

Bibliography

[1] Akyildiz, I.F., 1988. "On the Exact and Approximate Throughput Analysis of Closed Queueing Networks with Blocking", *IEEE Transactions on Software Engineering* **14**, 62.

[2] Buzacott, J.A., 1989. "Queueing Models of Kanban and MRP Controlled Production Systems", *Engineering Costs and Production Economics* **17**, 3.

[3] Buzacott, J.A., S.M. Price, and J.G. Shanthikumar, 1992. "The Performance of Kanban Controlled Serial Production Systems", to appear in Proceedings of US/German Conference on New Developments in OR in Production Planning and Control, June 1992, Hagen Germany.

[4] Deleersnyder, J., T.J. Hodgson, H. Muller, and P. O'Grady, 1989. "Kanban Pull Systems: An analytic approach", *Management Science* **35**, 1079.

[5] Dallery, Y., and S.B. Gershwin, 1992. "Manufacturing Flow Line Systems: A Review of Models and Analytical Results," *Queueing Systems, Theory and Applications* **12**, 3.

[6] Duenyas, I., and M. Keblis, 1993. "Release Policies for Assembly Systems", Technical Report 93-15, Department of Industrial and Operations Engineering, The University of Michigan.

[7] Hall, R.W., 1983. *Zero Inventories*, Dow Jones-Irwin, Homewood, IL.

[8] Karmarkar, U.S., and S. Kekre, 1989. "Batching Policy in Kanban Systems", *Journal of Manufacturing Systems*, **8**, 317.

[9] Kimura, O., and H Terada, 1981. "Design and Analysis of Pull Systems, a Method of Multi-stage Production Control", *International Journal of Production Research*, **19**, 241.

[10] Mitra, D., and I. Mitrani, 1990. "Analysis of a Kanban Discipline for Cell Coordination in Production Lines", *Management Science* **36**, 1548.

[11] Mitra, D., and I. Mitrani, 1991. "Analysis of a Kanban Discipline for Cell Coordination in Production Lines, II: Stochastic Demands", *Operations Research* **39**, 807.

[12] Monden, Y., 1983. *Toyota Production System*, Industrial Engineering and Management Press.

[13] Ohno, T., 1988. *Toyota Production System: Beyond Large Scale Production*, Productivity Press, Cambridge, MA (original Japanese 1978).

[14] Reiser, M., and S. Lavenberg, 1980. "Mean Value Analysis of Closed MultiChain Queueing Networks", *J.ACM* **27**, 313.

[15] Shanthikumar, J., and M. Gocmen, 1983. "Heuristic Analysis of Closed Queueing Networks", *International Journal of Production Research* **21**, 675.

[16] Spearman, M.L., and 1992. "Customer Service in Pull Systems", *Operations Research* **40**, 948.

[17] Spearman, M.L., D.L. Woodruff, and W.J. Hopp, 1990. "CONWIP: A Pull Alternative to Kanban", *International Journal of Production Research* **28**, 879.

[18] Spearman, M.L., and M.A. Zazanis, 1992. "Push and Pull Production systems: Issues and Comparisons", *Operations Research* **40**, 521.

[19] Tayur, S.R., 1992. "Structural Properties and a Heuristic for Kanban Controlled Serial Lines", Technical Report, Graduate School of Industrial Administration, Carnegie-Mellon University, to appear in *Management Science*.

[20] Uzsoy, R, and L.A. Martin-Vega, 1990. "Modelling Kanban-based Demand-pull Systems: A Survey and Critique", *Manufacturing Review* **3**, 155.

[21] Whitt, W. 1984. "Open and Closed Models for Networks of Queues", *AT&T Technical Journal* **63**, 1911-1979